Problem B. Candy

Time limit 1000 ms
Mem limit 262144 kB

Not so long ago, Vlad had a birthday, for which he was presented with a package of candies. There were n types of candies, there are a_i candies of the type i ($1 \le i \le n$).

Vlad decided to eat exactly one candy every time, choosing any of the candies of a type that is currently the most frequent (if there are several such types, he can choose **any** of them). To get the maximum pleasure from eating, Vlad **does not want** to eat two candies of the same type in a row.

Help him figure out if he can eat all the candies without eating two identical candies in a row.

Input

The first line of input data contains an integer t ($1 \le t \le 10^4$) — the number of input test cases.

The following is a description of t test cases of input, two lines for each.

The first line of the case contains the single number n ($1 \le n \le 2 \cdot 10^5$) — the number of types of candies in the package.

The second line of the case contains n integers a_i ($1 \le a_i \le 10^9$) — the number of candies of the type i.

It is guaranteed that the sum of n for all cases does not exceed $2 \cdot 10^5$.

Output

Output t lines, each of which contains the answer to the corresponding test case of input. As an answer, output "`YES`" if Vlad can eat candy as planned, and "`NO`" otherwise.

You can output the answer in any case (for example, the strings "`yEs`", "`yes`", "`Yes`" and "`YES`" will be recognized as a positive answer).

Examples

Input	Output

6
2
2 3
1
2
5
1 6 2 4 3
4
2 2 2 1
3
1 1000000000 999999999
1
1
	YES
	NO
	NO
	YES
	YES
	YES

Note

In the first example, it is necessary to eat sweets in this order:

- a candy of the type 2, it is the most frequent, now $a = [2, 2]$;
- a candy of the type 1, there are the same number of candies of the type 2, but we just ate one, now $a = [1, 2]$;
- a candy of the type 2, it is the most frequent, now $a = [1, 1]$;
- a candy of the type 1, now $a = [0, 1]$;
- a candy of the type 2, now $a = [0, 0]$ and the candy has run out.

In the second example, all the candies are of the same type and it is impossible to eat them without eating two identical ones in a row.

In the third example, first of all, a candy of the type 2 will be eaten, after which this kind will remain the only kind that is the most frequent, and you will have to eat a candy of the type 2 again.